Welcome to the Digital Certificates Webcast

Michael Blaha (Michael.Blaha@Broadcom.com)
Katie Juhala (Katie.Juhala@Broadcom.com)

July 2021

Disclaimer

Certain information in this presentation may outline CA's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. This presentation is based on current information and resource allocations as of 15th October 2019 and is **subject to change or withdrawal by CA at any time without notice**. **The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion**.

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to CA maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. Broadcom assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will Broadcom be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if Broadcom is expressly advised in advance of the possibility of such damages.

Broadcom Digital Certificate Video Training Sessions

Web-based training is provided at no cost for customers on active maintenance.

Customers: To learn more about the training options and to take web-based training, visit Mainframe Education and click TRAINING LOGIN. After logging in, search by course name or course code.
Broadcom employees: Access Mainframe Education via your Learning@Broadcom tile.

Learning Path

All Roles

Course Name	Code	Type	Length
Digital Certificate Overview 200	06SEC20010	Video	15 Minutes
Client/Server Certificate Configuration and Authentication 200	06SEC20020	Video	10 Minutes
Keyring and Certificate Authorization 200	06SEC20030	Video	10 Minutes
Debugging - SSL Keyring/Certificate Problems 200	06SEC20040	Video	20 Minutes
Renew ing Certificates - Internal Certificate Authority 200	06SEC20050	Video	10 Minutes
Renew ing Certificates - External Certificate Authority 200	06SEC20060	Video	10 Minutes
CA SMP/E Internet Service Retrieval Configuration 200	06SEC20070	Video	15 Minutes

Broadcom Digital Certificate Video Training Sessions

Course Name	Code	Type	Length
Digital Certificate Overview 200	06SEC20010	Video	15 Minutes
Client/Server Certificate Configuration and Authentication 200	06SEC20020	Video	10 Minutes
Keyring and Certificate Authorization 200	06SEC20030	Video	10 Minutes
Debugging - SSL Keyring/Certificate Problems 200	06SEC20040	Video	20 Minutes
Renew ing Certificates - Internal Certificate Authority 200	06SEC20050	Video	10 Minutes
Renew ing Certificates - External Certificate Authority 200	06SEC20060	Video	10 Minutes
CA SMP/E Internet Service Retrieval Configuration 200	06SEC20070	Video	15 Minutes

Digital Certificates

- Summary
  - What is a digital certificate?
  - Common Types of Certificates
  - What is a Certificate's 'Chain of Trust'?
  - Types of SSL Client/Server Configurations
  - How is SSL Configured?
  - Example Configurations
  - How does SSL work between client and server?
  - Keyring Access
  - Certificate Private Key Access
  - Digital Certificate Administration Authorization
  - Renewing Certificates

Digital Certificates

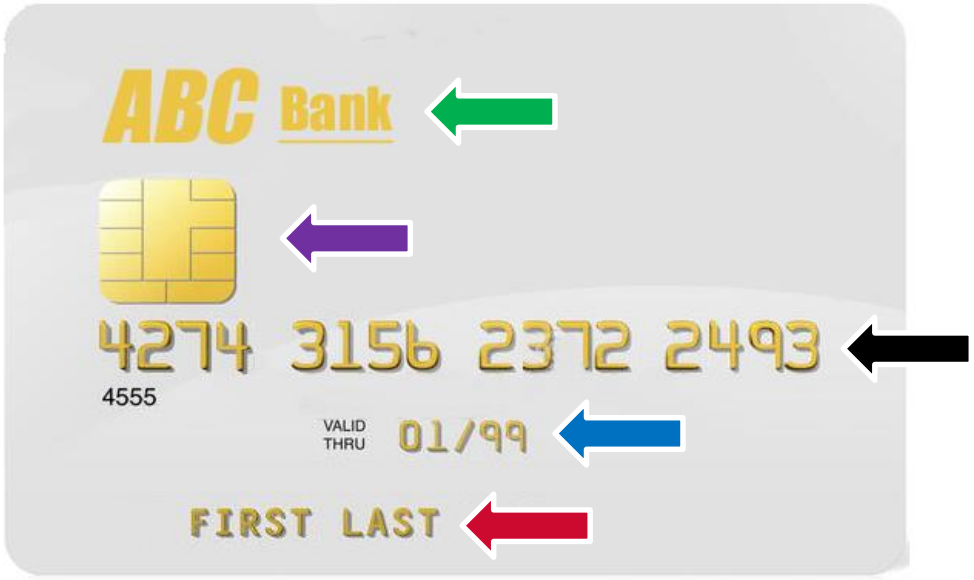
What is a digital certificate?

- Can be used as an alternative to requesting userid and password information, a z/OS Client or Server task can authenticate users based on their digital certificates.
- Digital certificates provide a means of authentication through the use of public-key cryptography and a trusted third party, known as a Certification Authority(CA).

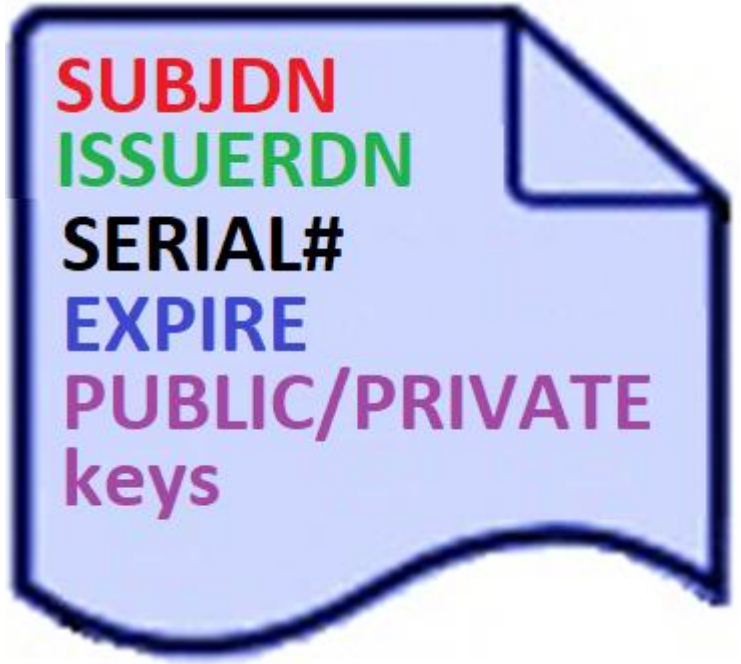
Digital Certificates

What is a digital certificate? continued

- A certificate is comparable to an identity card/credit card. It basically holds:



Credit Card



Certificate

Digital Certificates

What is a digital certificate? continued

Authentication

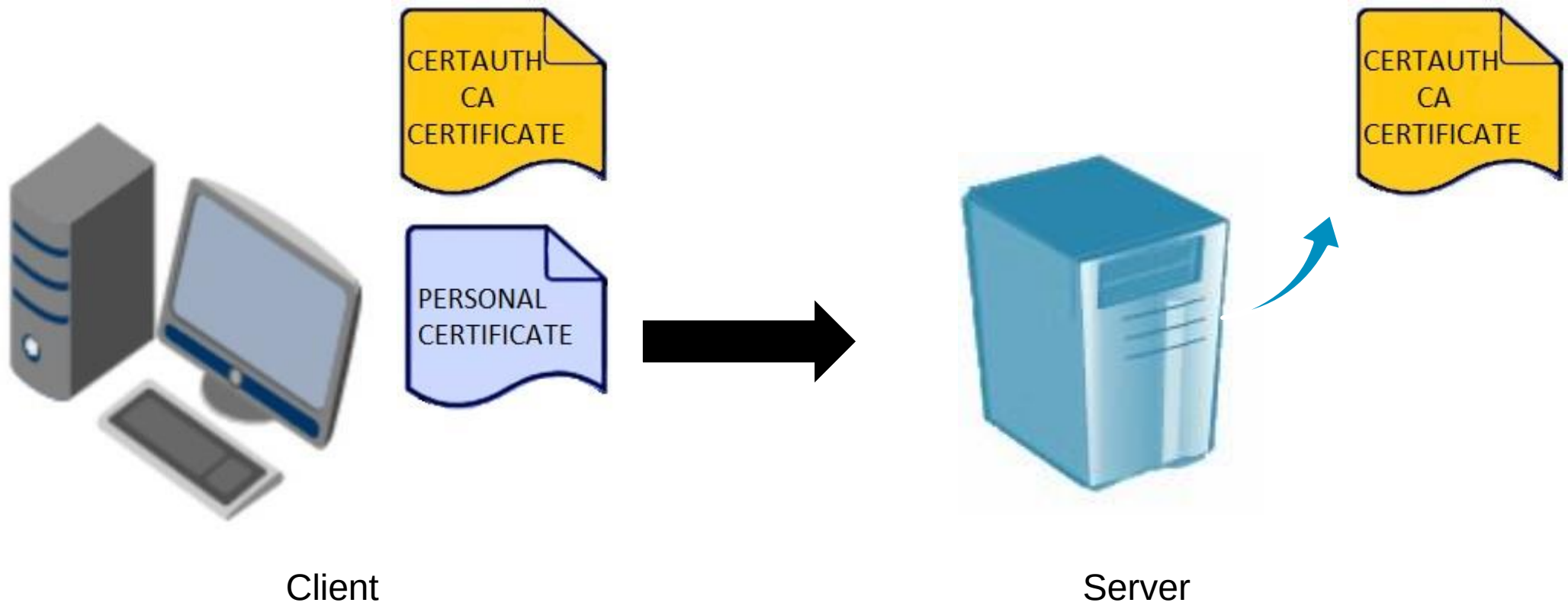
Credit Card

Vendor

Digital Certificates

What is a digital certificate? Continued

Sample Client Authentication



Digital Certificates

Common Types of Certificates

- Personal Certificates are associated with a userid and have a **private key**.
  - User: Specifies the userid that is to be associated with the certificate. This type of certificate usually has a Private Key and is also known as a PERSONAL certificate.
  - Server: Similar to a User certificate that is associated with a server task's userid, can be used for encrypting and decrypting the content. This type of certificate usually has a Private Key and is also known as a PERSONAL certificate.
  - Client: Similar to a User certificate that is associated with a client task's userid used for authenticating the client to the server. This type of certificate usually has a Private Key and is also known as a PERSONAL certificate.
  - SITECERT: SITECERT in place of a userid indicates that the certificate is a site certificate used to share a single certificate and its private key among multiple userids. This type of certificate is like a User certificate except that it is shared by multiple userids.

Digital Certificates

Common Types of Certificates

- CA certificate also known as a signer certificate or CERTAUTH certificate
  - CA certificate: The certificate is a Certification Authority certificate used to sign and to verify signatures of other certificates. A CA certificate can be an intermediate or a root signing certificate.

Digital Certificates

What is a Certificate's 'Chain of Trust'?

Certificate chain is made up of a list of certificates that start from a PERSONAL certificate and terminate with the root certificate.

Certificate Signing(authentication) Chain Example

- Personal certificate MyServer signed by a CA certificate INTER2.
- CA certificate INTER2 is signed by CA certificate INTER1.
- CA certificate INTER1 is signed by CA certificate USROOT.
- CA certificate USROOT is self signed.

Digital Certificates

What is a Certificate's 'Chain of Trust'? continued

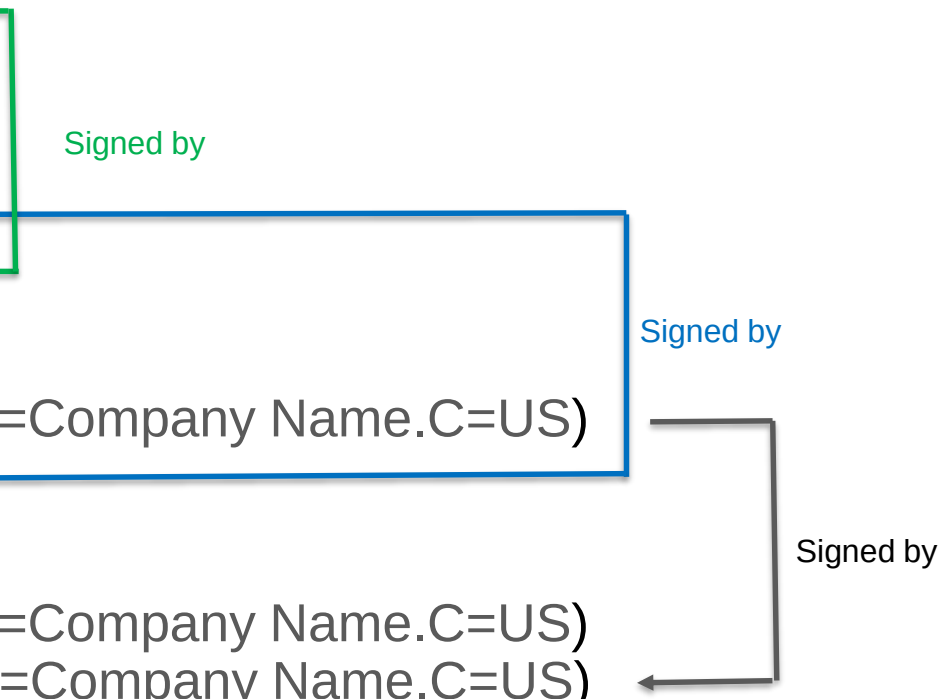
Example Certificate Signing Chain

MyServer.CERT
ISSUERDN(CN=Inter2Operations.OU=MyCo.C=US)
  SUBJDN(CN=MyServerOperations.OU=MyCo.C=US)

CERTAUTH.INTER2
ISSUERDN(CN=InterOperations.OU=MyCo.C=US)
  SUBJDN(CN=Inter2Operations.OU=MyCo.C=US)

CERTAUTH.INTER1
ISSUERDN(CN=USRootCert.OU=Auditing Department.O=Company Name.C=US)
  SUBJDN(CN=InterOperations.OU=MyCo.C=US)

CERTAUTH.USROOT
ISSUERDN(CN=USRootCert.OU=Auditing Department.O=Company Name.C=US)
  SUBJDN(CN=USRootCert.OU=Auditing Department.O=Company Name.C=US)



Digital Certificates

What is a Certificate's 'Chain of Trust'?

ACF2 GENCERT commands to create sample certificates.

```
ACF
SET PROFILE(USER) DIVISION(CERTDATA)
* GENCERT CA Root Certificate *
GENCERT CERTAUTH.USROOT SUBJ(CN='USRootCert' OU='MyCo' C=US)
  LABEL(US Root Cert)
* GENCERT CA Intermediate 2 Certificate *
GENCERT CERTAUTH.inter1 SUBJ(CN='InterOperations' OU='MyCo' C=US)
  LABEL(Intermediate One) SIGNWITH(certauth Label(US Root Cert))
* GENCERT CA Intermediate 2 Certificate *
GENCERT CERTAUTH.inter2 SUBJ(CN='Inter2Operations' OU='MyCo' C=US)
  LABEL(Intermediate Two) SIGNWITH(certauth Label(Intermediate One))
* GENCERT Personal User Certificate *
GENCERT USRTEST.cert SUBJ(CN='USRTESTOperations' OU='MyCo' C=US)
  LABEL(USRTEST User) SIGNWITH(certauth Label(Intermediate Two))
```


Digital Certificates

What is a Certificate's 'Chain of Trust'?

Top Secret GENCERT commands to create sample certificates.

```
 /* GENCERT CA Root Certificate */
TSS GENCERT(CERTAUTH) DIGICERT(USROOT) LABLCERT('US Root Cert') -
 SUBJECTN('CN="USRootCert" OU="MyCo" C=US')
 /* GENCERT CA Intermediate 1 Certificate */
TSS GENCERT(CERTAUTH) DIGICERT(INTER1) LABLCERT('Intermediate One') -
 SUBJECTN('CN="InterOperations" OU="MyCo" C=US') -
 SIGNWITH(certauth,USROOT)
 /* GENCERT CA Intermediate 2 Certificate */
TSS GENCERT(CERTAUTH) DIGICERT(INTER2) LABLCERT('Intermediate Two') -
 SUBJECTN('CN="Inter2Operations" OU="MyCo" C=US') -
 SIGNWITH(certauth,INTER1)
 /* GENCERT Personal User Certificate */
TSS GENCERT(USER002) DIGICERT(USRTEST) LABLCERT('USRTEST User') -
 SUBJECTN('CN="USRTEST" OU="MyCo" C=US') -
 SIGNWITH(certauth,INTER2)
```


Digital Certificates

What is a Certificate's 'Chain of Trust'?

RACF GENCERT commands to create sample certificates.

```
 /* GENCERT CA Root Certificate */
RACDCERT GENCERT CERTAUTH WITHLABEL('US Root Cert') +
 SUBJECTSDN(CN('USRootCert') OU('MyCo') C('US')) +
 /* GENCERT CA Intermediate 1 Certificate */
RACDCERT GENCERT CERTAUTH WITHLABEL('Intermediate One') +
 SUBJECTSDN(CN('InterOperations') OU('MyCo') C('US')) +
  SIGNWITH(CERTAUTH LABEL('US Root Cert'))
 /* GENCERT CA Intermediate 2 Certificate */
RACDCERT GENCERT CERTAUTH WITHLABEL('Intermediate Two') +
 SUBJECTSDN(CN('Inter2Operations') OU('MyCo') C('US')) +
 SIGNWITH(CERTAUTH LABEL('Intermediate One'))
 /* GENCERT Personal User Certificate */
RACDCERT GENCERT ID(USER002) WITHLABEL('USRTEST User') +
 SUBJECTSDN(CN('USRTESTOperations') OU('MyCo') C('US')) +
 SIGNWITH(CERTAUTH LABEL('Intermediate Two'))
```


Digital Certificates

What is a Certificate's 'Chain of Trust'?

ACF2 verify the signing chain of a personal certificate.

Command:

ACF
CHKCERT *USRTEST.cert* **CHAIN**

Results:

.. Certificate information ..

Chain Information:

Chain contains 4 certificates

Chain is COMPLETE

Digital Certificates

What is a digital certificate?

Top Secret verify the signing chain of a personal certificate.

Command:

```
TSS EXPORT(USER002) DIGICERT(USRTEST) DCDSN('USER002.CERTTOM') -
   FORMAT(PKCS7DER)
TSS CHKCERT ('USER002.CERTTOM') CHAIN
```

* USER002 is the ACID that owns the Personal Certificate

Results:

```
.. Certificate information ..

Chain Information:
    Chain contains    4 certificates
    Chain is complete
TSS0300I  CHKCERT  FUNCTION SUCCESSFUL
```


Digital Certificates

What is a digital certificate?

RACF verify the signing chain of a personal certificate.

Command:

RACDCERT LISTCHAIN(LABEL('*USRTEST User*'))

Results:

.. Certificate information ..

Chain information:

Chain contains 4 certificate(s), chain is complete

Chain contains no ring in common

Client/Server Configuration and Keyrings/Keystores

- Summary
  - Types of SSL Client/Server Configurations
  - How is SSL Configured?
  - Example Configurations
  - How does SSL work between client and server?

Client/Server Configuration and Keyrings/Keystores

Types of SSL Client/Server Configurations

- What is SSL?
- For SSL Client/Server Authentication support there are two types of configurations.
  - Server authentication: This is the usual SSL setup where the server sends it's server personal certificate to the client for authentication.
  - Server and client authentication: This is not as common but occasionally done. The server sends it's server personal certificate to the client for authentication and the client sends it's client personal certificate to the server for authentication.

Client/Server Configuration and Keyrings/Keystores

Client/Server Configuration and Keyrings/Keystores

- Sample Client/Server configurations:
  - Both the Client and Server on the same z/OS LPAR.
  - The Client and Server on different z/OS LPARs.
  - The Client is on a z/OS LPAR and the server is on a Linux, UNIX, or Windows platform.
  - The Client is on a Linux, UNIX, or Windows platform and the Server is on z/OS LPAR.
  - Other combinations of platforms and environments.
- z/OS Keyrings
- Linux, UNIX, or Windows Keystores.

Client/Server Configuration and Keyrings/Keystores

Certificate Configuration for Authentication

- What certificates are needed for Server Authentication?
  - The Server requires two certificates.
  - The Client requires one certificate.
- What certificates are needed for Server and Client Authentication?
  - The Server requires three certificates.
  - The Client requires three certificates.
- How are the certificates configured?
  - Identify the Client and the Server.
  - Deploy the required Client and Server certificates.
  - Add the required Client and Server certificates to the appropriate Keyring or Keystore.

Client/Server Configuration and Keyrings/Keystores

Client/Server Configuration and Keyrings/Keystores

Server authentication Windows Client, z/OS Server:

Windows Client A Keystore

- Server B CERTAUTH signer

z/OS Server B Keyring

- Server B CERTAUTH signer
- Server B PERSONAL cert with private key

Client/Server Configuration and Keyrings/Keystores

Client/Server Configuration and Keyrings/Keystores

Server authentication z/OS Client, Linux Server:

z/OS Client A Keyring

- Server B CERTAUTH signer

Linux Server B Keystore

- Server B CERTAUTH signer
- Server B PERSONAL cert with private key

Client/Server Configuration and Keyrings/Keystores

Client/Server Configuration and Keyrings/Keystores

Server authentication z/OS Client, z/OS Server:

z/OS Client A Keyring

- Server B CERTAUTH signer

z/OS Server B Keyring

- Server B CERTAUTH signer
- Server B PERSONAL cert with private key

Client/Server Configuration and Keyrings/Keystores

Keyring/Certificate Server Authentication Overview

Example 1: Keyring/Certificate setup For Server authentication:

Client A Keyring

- Server B CERTAUTH signer

Server B Keyring

- Server B CERTAUTH signer
- Server B PERSONAL cert with private key

Server Authentication **Overview**

1. Client initiates session with Server.
2. Server sends Personal Certificate.
3. Client authenticates Server Personal Certificate and establishes a secured connection.

Client/Server Configuration and Keyrings/Keystores

How are Certificates and Keyrings are used in Client/Server Authentication? continued

Example 1: Keyring/Certificate Server Authentication **Details**:

1. Client connects to a server and requests that the server identify itself.
2. Server sends its personal Certificate with a public key.
3. Client verifies the server's personal certificate. If the client trusts the certificate, a symmetric session key is sent back to the server.
4. Server decrypts the symmetric session key and sends back an acknowledgement with the session key.
5. Server and Client now encrypt all transmitted data with the session key.

Client/Server Configuration and Keyrings/Keystores

Keyring/Certificate Client and Server Authentication

Example 2: Keyring/Certificate setup For Client/Server authentication :

Client A Keyring

- Server B CERTAUTH signer
- Client A PERSONAL CERTAUTH signer
- Client A PERSONAL cert with private key

Server B Keyring

- Server B CERTAUTH signer
- Server B PERSONAL cert with private key
- Client A PERSONAL CERTAUTH signer

Client and Server Authentication **Overview**

1. After Server is authenticated, Client sends its Personal Certificate.
2. Server authenticates Client Personal Certificate and establishes a secured connection.

Client/Server Configuration and Keyrings/Keystores

How are Certificates and Keyrings are used in Client/Server Authentication? continued

Example 2: Keyring/Certificate Client/Server Authentication **Details** :

1. Client connects to a server and requests that the server identify itself.
2. Server sends its personal Certificate with a public key.
3. Client verifies the server's personal certificate. If the client trusts the certificate, a symmetric session key is sent back to the server along with it's personal certificate.
4. Server verifies the client's personal certificate. If the client can be trusted, the server decrypts the symmetric session key using its private key and Server decrypts the symmetric session key and sends back an acknowledgement with the session key.
5. Server and Client now encrypt all transmitted data with the session key.

Keyring and Certificate Security

Summary

- Keyring Access
- Certificate Private Key Access
- Digital Certificate Administration Authorization

Keyring and Certificate Security

Keyring Access

- Resource checks for a client or server task access to a Keyring.
- These resource checks are driven by USS R_datalib calls.
- Two resources checked, if the first check fails, the second check is done.

1. ***Ring-specific profile checking***

   Resource Class: RDATALIB * ACF2 default TYPE(RDA)
   Resource: <ringOwner>.<ringName>.LST
   Access: **READ**

2. ***Global profile checking***

   Resource Class: FACILITY * ACF2 default TYPE(FAC)
   Resource: IRR.DIGTCERT.LISTRING
   Access: **READ** allows access to key ring that owned* by the user's own userid.
   UPDATE allows access to a key ring that is owned* by another user's userid.

* Keyring Ownership, sample ESM Create Keyring commands, FTPD is the owner:

ACF2: INSERT *FTPD*.ftpdring RINGNAME(FTPDringname)
TOP Secret: TSS ADD(*FTPD*) KEYRING(FTPDRING) LABLRING(FTPDringname)
RACF: RACDCERT ID(*FTPD*) ADDRING(FTPDringname)

Keyring and Certificate Security

Keyring Access **ACF2** Example

Keyring access for FTPD Server Task Userid

- Resource rule for ***Ring-specific profile checking***

  ACF
  SET RESOURCE(RDA)
  RECKEY <ringOwner> ADD(<ringName>.LST USER(FTPD) SERVICE(READ) ALLOW)

- Resource rule for ***Global profile checking***

  * Allow access to a keyring owned by FTPD
  ACF
  SET RESOURCE(FAC)
  RECKEY IRR ADD(DIGTCERT.LISTRING USER(FTPD) SERVICE(READ) ALLOW)

  * Allow access to a Keyring owned by another user's userid
  ACF
  SET RESOURCE(FAC)
  RECKEY IRR ADD(DIGTCERT.LISTRING USER(FTPD) SERVICE(UPDATE) ALLOW)

Keyring and Certificate Security

Keyring Access **Top Secret** Example

Keyring access for FTPD Server Task Userid

- Resource rule for ***Ring-specific profile checking***

  TSS PERMIT(FTPD) RDATALIB(<ringOwner>.<ringName>.LST) ACCESS(READ)

- Resource rule for ***Global profile checking***

  * Allow access to a keyring owned by FTPD
  TSS PERMIT(FTPD) IBMFAC(IRR.DIGTCERT. LISTRING) ACCESS(READ)

  * Allow access to a Keyring owned by another user's userid
  TSS PERMIT(FTPD) IBMFAC(IRR.DIGTCERT. LISTRING) ACCESS(UPDATE)

Keyring and Certificate Security

Keyring Access **RACF** Example

Keyring access for FTPD Server Task Userid

- Resource rule for ***Ring-specific profile checking***

  PERMIT <ringOwner>.<ringName>.LST CLASS(RDATALIB) ID(FTPD) ACCESS(READ)

- Resource rule for ***Global profile checking***

  * Allow access to a keyring owned by FTPD
  PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(FTPD) ACCESS(READ)

  * Allow access to a Keyring owned by another user's userid
  PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(FTPD) ACCESS(UPDATE)

Keyring and Certificate Security

Certificate Private Key Access

Requirements for a Client or Server Task userid access to the private key of a Personal or SITECERT certificate:

1. The certificate is connected to it's keyring with the PERSONAL usage option.

2. One of the following conditions is true for the Client or Server's Task Userid:

   a) The userid is the owner of the certificate.

   b) The caller's userid has access to:
   Resource Class: RDATALILB * ACF2 default TYPE(RDA)
   Resource: <ringOwner>.<ringName>.LST
   Access: **UPDATE**

   c) For a SITECERT certificate, the caller's userid has access to:
   Resource Class: FACILITY * ACF2 default TYPE(FAC)
   Resource: IRR.DIGTCERT.GENCERT
   Access: **DELETE**

Keyring and Certificate Security

Certificate Private Key Access **ACF2** Example

Certificate Private Key access for FTPD Server Task Userid

- ***Ownership***
  Sample ACF2 GENCERT, FTPD is the owner:

  GENCERT *FTPD*.CERT SUBJ(CN='FTPd Server Certificate')

- Resource rule for Private Key of a Personal certificate not owned by FTPD:

  ACF
  SET RESOURCE(RDA)
  RECKEY <ringOwner> ADD(<ringName>.LST USER(FTPD) SERVICE(UPDATE) ALLOW)

- Resource rule for Private Key of a SITECERT certificate:

  ACF
  SET RESOURCE(FAC)
  RECKEY IRR ADD(DIGTCERT.GENCERT USER(FTPD) SERVICE(DELETE) ALLOW)

Keyring and Certificate Security

Certificate Private Key Access **TOP SECRET** Example

Certificate Private Key access for FTPD Server Task Userid

- ***Ownership***
  Sample TOP SECRET GENCERT, FTPD is the owner:

  TSS GENCERT(*FTPD*) DIGICERT(FTPSCERT) SUBJECTN('CN="FTPd Server Certificate"')

- Resource rule for Private Key of a Personal certificate not owned by FTPD:

  TSS PERMIT(FTPD) RDATALIB(<ringOwner>.<ringName>.LST) ACCESS(UPDATE)

- Resource rule for Private Key of a SITECERT certificate:

  TSS PERMIT(FTPD) IBMFAC(IRR.DIGTCERT.GENCERT) ACCESS(CONTROL)

Keyring and Certificate Security

Certificate Private Key Access **RACF** Example

Certificate Private Key access for FTPD Server Task Userid

- ***Ownership***
  Sample RACF GENCERT, FTPD is the owner:

  RACDCERT GENCERT ID(*FTPD*) SUBJECTSDN(CN('FTPd Server Certificate')
- Resource rule for Private Key of a Personal certificate not owned by FTPD:

  PERMIT <ringOwner>.<ringName>.LST CLASS(RDATALIB) ID(FTPD) ACCESS(UPDATE)
- Resource rule for Private Key of a SITECERT certificate:

  PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(FTPD) ACCESS(CONTROL)

Keyring and Certificate Security

Digital Certificate Administration Authorization

- Special Privileges
  - ACF2: Security privilege
  - Top Secret: MSCA or SCA security level
  - RACF: SPECIAL attribute

Keyring and Certificate Security

Digital Certificate Administration Authorization continued

- Granular Certificate Administration Authority Resource Checks
  - ACF2:
    Resource Class: CASECAUT * ACF2 default TYPE(AUT)
    Resource: ACFCMD.USER.cmd
    Access: *varies on ownership and the command*
  - Top Secret:
    Resource Class: CASECAUT
    Resource: TSSCMD.USER.cmd
    Access: *varies on ownership and the command*
  - RACF:
    Resource Class: FACILITY
    Resource: IRR.DIGTCERT.function
    Access: *varies on ownership and the command*

Renewing Digital Certificates

- Summary
  - What is a Certificate Authority
    - Internal Certificate Authority
    - External Certificate Authority
  - How to identify certificate expiration dates with ACF2, Top Secret and RACF
  - Instructions for renewing Certificate Authority signed certificates
    - ACF2
    - Top Secret
    - RACF

Renewing Digital Certificates

What is a Certificate Authority(CA)?

- An entity that issues and verifies digital certificates
- Can be internal or external to the organization in need of a digital certificate
- What is an Internal Certificate Authority (CA)?
  - An Internal CA, also known as a local CA, is when a site acts as their own CA and creates(GENCERT) and signs their own certificates which includes the ROOT and INTERMEDIATE certificates.
- What is an External Certificate Authority (CA)?
  - External CAs are trusted third parties(Examples Verisign, GoDaddy, Entrust, Symantec, Thawte, Comodo, SecureNet etc.) which sign certificates for other sites. External CA's root certificates (containing their public keys) are bundled in popular web browsers.

Renewing Digital Certificates

How to check for expiring certificates – ACF2 and Top Secret

- ACF2 and Top Secret use the SAFCRRPT Certificate Utility
- The certificate utility displays the certificate hierarchy in your database
- The output can be tailored to display certificates that will expire within a specified number of days

The following is sample JCL to run the certificate utility. This JCL is found in the CAX1JCL library (ACF2) or CAKOJCL library (Top Secret). The member name is CERTUTIL:

```
//EXPIRE   EXEC PGM=SAFCRRPT,PARM='TITLE(Certificate Expiration Report)'
//SYSUDUMP   DD   SYSOUT=*
//SYSPRINT   DD   SYSOUT=*
//SYSIN      DD   *
RECORDID(-)
EDAYS(60)
FIELDS(EXPIRE)
```


Renewing Digital Certificates

SAFCRRPT Certificate Utility Output

```
********************************* TOP OF DATA ***********************************
CA Mainframe Security     - CERTIFICATE UTILITY REPORT          - PAGE    1
DATE 02/24/21 (21.055) TIME 11.37

Report Parameters:
     TITLE(CERTIFICATE UTILITY REPORT) RECORDID(-) EDAYS(60) FIELDS(EXPIRE)


Record id - CERTAUTH.DOWNLOAD          Signed by:  CERTAUTH.DGCERTI
          Expire Date     2021/04/08

Record id - CERTAUTH.TEST                         Signed by:  None - Self-Signed
          Expire Date     2021/04/25

Record id - USER01.CERT                           Signed by:  None - Self-Signed
          Expire Date     2021/04/08

Record id - USERTEST.CERT                         Signed by:  CERTAUTH.TEST
          Expire Date     2021/04/25
```


Renewing Digital Certificates

SAFCRRPT Certificate Utility Output (Continued)

```
CA Mainframe Security     - CERTIFICATE UTILITY REPORT          - PAGE    2
DATE 02/24/21 (21.055) TIME 11.37
          Total Certificates                    04
          CA Certificates                       02
          Site Certificates                     00
          User Certificates                     02
          Expired Certificates                  00
          Inactive Certificates                 00
          ICSF Certificates                     00
          PCICC Certificates                    00
          Self-signed certificates              02
          RSA certificates                      04
          DSA certificates                      00
          ECC certificates                      00
          DH certificates                       00
          Trusted Certificates                  04
          High Trust Certificates               00
******************************** BOTTOM OF DATA ********************************
```


Renewing Digital Certificates

How to check for expiring certificates – RACF

- RACF utilizes IBM Health Checker RACF checks (IBMRACF)
- DAYS parameter will give control to see certificates expiring within a range
- For more information regarding the RACF_CERTIFICATE_EXPIRATION check, please see the IBM documentation: https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.e0zl100/racfcertificate.htm

Renewing Digital Certificates – Internal CA

How to renew internal certificates – The RENEW Command

Before using the RENEW command:

- Where does the certificate exist?
- Does the certificate have a private key?
- Is there a signing certificate?
  - Where does it exist?
  - When does it expire?
  - Does it have a private key?
  - Should I use the SIGNWITH parameter?

Renewing Digital Certificates – Internal CA

How to renew internal certificates – ACF2

```
ACF
CHKCERT USER1.CERT
RENEW USER1.CERT EXPIRE(12/31/2030)
CHKCERT USER1.CERT
```

Before	After
Not valid before: 2021/02/26 00:00:00 UTC Not valid after: 2022/02/26 23:59:59 UTC	Not valid before: 2021/02/26 00:00:00 UTC Not valid after: 2030/12/30 23:59:59 UTC

Additional parameters can be found in ACF2 Documentation under Digital Certificate Support

Renewing Digital Certificates – Internal CA

How to renew internal certificates – Top Secret

```
TSS EXPORT(USER1) DIGICERT(USR1TEST) -
 DCDSN('USER1.CERTTOM.BEFORE') FORMAT(PKCS7DER)
TSS CHKCERT DCDSN(USER1.CERTTOM.BEFORE')
TSS RENEW(USER1) DIGICERT(USR1TEST) NADATE(12/30/30)
TSS EXPORT(USER1) DIGICERT(USR1TEST) -
 DCDSN('USER1.CERTTOM.AFTER') FORMAT(PKCS7DER)
TSS CHKCERT DCDSN('USER1.CERTTOM.AFTER')
```

Before	After
NOT BEFORE = 2021/03/04 00:00:00 UTC NOT AFTER = 2022/03/04 23:59:59 UTC	NOT BEFORE = 2021/03/11 00:00:00 UTC NOT AFTER = 2030/12/30 23:59:59 UTC

Additional parameters can be found in TSS Documentation under RENEW Function

Renewing Digital Certificates – Internal CA

How to renew internal certificates – RACF

```
RACDCERT LIST(LABEL('USER1 Cert'))
RACDCERT ID(USER1) GENREQ(LABEL('USER1 Cert')) +
 DSN('SYSADM.CERT.REQ')
RACDCERT ID(USER1) GENCERT('SYSADM.CERT.REQ') +
 SIGNWITH(CERTAUTH LABEL('Intermediate Test')) +
 NOTAFTER(DATE(2030-12-30))
RACDCERT LIST(LABEL('USER1 Cert'))
```

Before	After
Start Date: 2021/03/18 00:00:00 End Date: 2022/03/18 23:59:59	Start Date: 2021/03/18 00:00:00 End Date: 2030/12/30 23:59:59

This process can be found in the IBM documentation under Renewing an expiring certificate

Renewing Digital Certificates – External CA

External CA Renewal Process

1. Site creates a Certificate Signing Request (CSR) for the expiring certificate.
2. Site sends the CSR to the External CA for signing.
3. External CA signs and returns the signed certificate.
4. Site inserts the signed certificate from the External CA replacing the original certificate that was GENREQed.

Renewing Digital Certificates – External CA

External CA Renewal Process

Tip: EXPORT the certificate to a dataset to save it – just in case

- If the private key is non-ICSF, use PKCS#12 format to save the certificate and its public/private key pair.
- If the private key is ICSF, consider using the IBM freeware utility called KEYXFER to backup the private key in conjunction with a non-PKCS#12 format (CERTDER) to backup the certificate and public key.

```
ACF2: EXPORT user1.cert DSN('USER1.CERT.SAVED') FORMAT(PKCS12DER) PASSWORD(pkcs12 password)

TOP SECRET: TSS EXPORT(USER1) DIGICERT(USR1TEST) DCDSN('USER1.CERT.SAVED')
            FORMAT(PKCS12DER) PKCSPASS(pkcs12 password)

RACF: RADCERT EXPORT(LABEL('USER1 Cert')) DSN('USER1.CERT.SAVED') FORMAT(PKCS12DER)
      PASSWORD('pkcs12 password')
```


Renewing Digital Certificates – External CA

How to renew external certificates – ACF2

```
ACF
GENREQ USER1.CERT DSN('user1.cert.unsigned')

- Send to Certification Authority -

CHKCERT DSN('user1.cert.renewed')
SET PROFILE(USER) DIV(CERTDATA)
INSERT USER1.CERT DSN('user1.cert.renewed')TRUST
CHKCERT USER1.CERT
```

Before	After
Not valid before: 2021/02/26 00:00:00 UTC Not valid after: 2022/02/26 23:59:59 UTC	Not valid before: 2021/02/26 00:00:00 UTC Not valid after: 2030/12/30 23:59:59 UTC

Renewing Digital Certificates – External CA

How to renew external certificates – Top Secret – Part 1

```
TSS GENREQ(USER1) DIGICERT(OLDCERT) -
 DCDSN('USER1.CERT.UNSIGNED')

- Send to Certification Authority -

TSS CHKCERT DCDSN('USER1.CERT.RENEWED')
TSS ADD(USER1) DIGICERT(NEWCERT) DCDSN('USER1.CERT.RENEWED')
TSS EXPORT(USER1) DIGICERT(NEWCERT) -
 DCDSN('USER1.CERT.SAVED')FORMAT(PKCS7DER)
TSS CHKCERT DCDSN('USER1.CERT.SAVED')
```

Before	After
NOT BEFORE = 2021/03/04 00:00:00 UTC NOT AFTER = 2022/03/04 23:59:59 UTC	NOT BEFORE = 2021/03/11 00:00:00 UTC NOT AFTER = 2030/12/30 23:59:59 UTC

Renewing Digital Certificates – External CA

How to renew external certificates – Top Secret – Part 2

```
TSS EXPORT(USER1) DIGICERT(OLDCERT) -
 FORMAT(PKCS12DER) PKCSPASS(password) -
 DCDSN(OLDCERT.DIGICERT.DATASET)

TSS REM(USER1) DIGICERT(OLDCERT)

TSS EXPORT(USER1) DIGICERT(NEWCERT) -
 FORMAT(PKCS12DER) PKCSPASS(password) -
 DCDSN(NEWCERT.DIGICERT.DATASET)

TSS ADD(USER1) DIGICERT(OLDCERT) FORMAT(PKCS12DER) -
 PKCSPASS(password) DCDSN(NEWCERT.DIGICERT.DATASET)

TSS ADD(USER1) KEYRING(TESTRING) RINGDATA(USER1,OLDCERT) -
 USAGE(PERSONAL)
```


Renewing Digital Certificates – External CA

How to renew external certificates – RACF

```
RACDCERT LIST(LABEL('USER1 Cert'))
RACDCERT ID(USER1) GENREQ(LABEL('USER1 Cert')) +
 DSN('SYSADM.CERT.REQ')
- Send to Certification Authority -
RACDCERT ID(USER1) ADD('SYSADM.CERT.SIGNED')
RACDCERT LIST(LABEL('USER1 Cert'))
```

Before	After
Start Date: 2021/03/18 00:00:00 End Date: 2022/03/18 23:59:59	Start Date: 2021/03/18 00:00:00 End Date: 2030/12/30 23:59:59

This process can be found in the IBM documentation under Renewing an expiring certificate

Q & A

- Are there any SSL client/server configuration issues or questions that you would like to discuss?
- Has your site encountered any certificate renewal issues that have not been addressed?
- Are there any questions on the authorization required for an application to access a keyring and certificate's private key?

Digital Certificates Webcast

July 2021

Thank You

The learning path for this digital certificate training is at:

https://community.broadcom.com/education/viewdocument/digital-certificates-learning-path?CommunityKey=bd92ecf3-d291-44ae-87ef-f17f7697397e

Michael Blaha (Michael.Blaha@Broadcom.com)
Katie Juhala (Katie.Juhala@Broadcom.com)


BROADCOM®
connecting everything®